Theoretical Computer Science 85 (1991) 213–229
Elsevier

Resolution for some first-order modal systems*

Marta Cialdea**

Université Paul Sabatier, 118 Zonte de Narbonne Toulouse, France

Communicated by G Jäger
Received January 1988
Revised August 1989

Abstract

M. Cialdea, Resolution for some first-order modal systems, Theoretical Computer Science 85 (1991) 213–229.

In this paper we present a resolution method for first-order modal logic, which derives from an extension of already known propositional methods. The main feature of our approach to the problem of quantification in modal logic is a purely syntactical solution, embedded in a propositional method, which is a very close relative of classical resolution. Such results represent a contribution to the extension of logic programming to intensional languages.

1. Introduction

In the last years nonclassical logics have shown to be a suitable tool for the formalization of several problems in important fields of computer science and artificial intelligence. In particular, modal logics present a broad spectrum of applications, allowing to express nonextensional concepts such as necessity, obligation (deontic logics), temporal reference (temporal logics), knowledge (epistemic logics). These logics have been used for the treatment of problems in fields such as problem solving [22], knowledge representation [18], natural language processing [22], information retrieval [29], analysis and synthesis of programs [24–27] and programming languages [16].

In spite of their expressive power, however, intensional logics are sometimes avoided, because of the lack of efficient proof procedures for them, and preference is given to classical logic. Therefore, an important question to be answered is how to extend, to intensional logics, the most efficient automated deduction results known for

* This paper contains a part of the results obtained in [8], with some improvement and generalization of the proof methods.

** Present affiliation: Dipartimento di Informatica e Sistemistica, Università di Roma, via Salaria 113, 00198 Roma, Italy.

classical logic and, in particular, the resolution method [7, 30], on which the programming language Prolog is based.

Automatizable proof procedures for quantified modal logics have been defined in [2, 3, 19, 21, 23].

Our approach represents a first-order extension of a resolution method, defined for several propositional modal systems [12–15]. Its main features are the possibility of an early detection and elimination of elementary inconsistencies, and its verified suitability as a basis for the implementation of a modal extension of Prolog, the programming language Molog [16], which is at present under development. Its first interpreter has been implemented in Prolog on a DPS-8 [4], and modalities for the realization of a compiled version are at present being studied [6].

The resolution method thereby used are, however, limited to act on prenex formulae, thus failing to capture, for example, an epistemic statement such as "The wise man knows that someone has a white hat".

Part of the results exposed in this work, namely, a Herbrand theorem analog, has appeared in print before [11]. This paper offers the opportunity to set that work in the context of automatic proof procedures for modal systems, showing how it can be used to extend existing propositional resolution methods to full first-order logic.

Section 2 contains the definition of the modal systems which are considered. Unlike [2], first-order resolution is defined here for systems which do not have the Barcan formula as an axiom. In this paper definitions and proofs are given in detail only for the case of one of the simplest modal systems, *D*, which is a subtheory of the better known modal systems *T*, *S4* and *S5* [20]. The paper includes, however, also some hints on the possibility of extending the same results to the systems *T* and *S4*. Therefore, in Section 2 we define not only *D*, but also *T* and *S4*. All the proofs presented in this paper are purely syntactical, so the definition of the semantics of the modal systems considered is not included. It can be found in any introductory text on modal logic, such as [20].

Section 3 contains some definitions, leading finally to the notion of modal skolemization. Successively, modal unification (Section 4) and modal resolution (Section 5) are defined. Section 6 contains an outline of the completeness and soundness proof for the resolution system. Finally (Sections 7 and 8), we give a brief presentation of different approaches to the extension of classical resolution to modal logic and compare such works with ours.

2. The modal systems under consideration

The logical language of modal systems consists of all the classical logical symbols and the modal operator $\Box$ (necessity). Modal formulae include all classical formulae and the formulae of the form $\Box A$, where A is a modal formula. The modal operator $\Diamond$ (possibility) is defined as follows:

$$\Diamond A \equiv \neg\Box\neg A$$

The three modal systems D, T and $S4$ are defined as follows. Their derivation rules are the classical rules and the necessitation rule:

$$\frac{A}{\Box A}.$$

The axioms of D are all the axioms of classical logic and

$(D1)$ $\Box(A \to B) \to (\Box A \to \Box B)$,

$(D2)$ $\Box A \to \Diamond A$.

The axioms of T are $(D1)$ and

(T) $\Box A \to A$.

The axioms of $S4$ are $(D1)$, (T) and

$(S4)$ $\Box A \to \Box\Box A$.

We note again that Barcan formula

$$\forall x \Box A(x) \to \Box \forall x A(x)$$

is not an axiom (nor a theorem) of any of the three theories. Thus, from a semantical point of view, the domains associated with possible worlds are not necessarily identical. However, if a world w is accessible from w', then the domain associated with w includes the domain associated with w'. The behaviour of the quantifiers w.r.t. modal operators can be described by observing that the following formulae are all theorems of the theories:

$$\Box \forall x A(x) \to \forall x \Box A(x),$$

$$\Diamond \forall x A(x) \to \forall x \Diamond A(x),$$

$$\exists x \Box A(x) \to \Box \exists x A(x),$$

$$\exists x \Diamond A(x) \to \Diamond \exists x A(x),$$

while their converses are not theorems:

$$\forall x \Box A(x) \to \Box \forall x A(x),$$

$$\forall x \Diamond A(x) \to \Diamond \forall x A(x),$$

$$\Box \exists x A(x) \to \exists x \Box A(x),$$

$$\Diamond \exists x A(x) \to \exists x \Diamond A(x).$$

These syntactical laws, i.e. the provability or nonprovability of the above formulae, can be seen as stating the following informal property: universal quantifiers can be moved outwards across modal operators, but not inwards; dually, existential quantifiers can be moved inwards across modal operators, but not outwards. The basic principle underlying our definition of modal unification will practically stem out of this simple feature.

3. Modal skolemization

The application of classical skolemization to modal formulae is not sound in general. For example, the set

$$\{\forall x \Diamond p(x), \Box \exists y \neg p(y)\}$$

can easily be seen to be consistent, while

$$\{\forall x \Diamond p(x), \Box \neg p(c)\},$$

where c is an individual constant, is an inconsistent set of formulae.

In this section we define a modal extension of the skolemization rule, which, by keeping track of the modal context in which a quantifier occurs, allows a sound definition of substitution for modally skolemized formulae.

From now onwards we consider $\Diamond$ as a primitive symbol.

3.1. If A is a subformula of a modal formula C, then the *modal degree* of A is equal to the number of modal operators which have A in their scope. Furthermore, if A has the form $QxB(x)$, where Q is a quantifier, then the modal degree of the variable x in C and the modal degree of the quantifier Q in C are both equal to the modal degree of A in C. The notion of modal degree is extended to apply to free variables by stipulating that if the variable x is free in C, then its modal degree is 0.

3.2. A modal formula A is a *positive formula* iff A contains no implications, and no logical symbol in A lies in the scope of a negation.

Clearly, every formula A can be transformed into an equivalent formula A' which is a positive formula by applying the classical equivalences and

$$\neg \Box A \equiv \Diamond \neg A,$$

$$\neg \Diamond A \equiv \Box \neg A.$$

Before introducing the modal notion of Skolem normal form, we define a relativized version of the classical notion of prenex formula. As there is no risk of confusion, the same names, "prenex normal form" and "Skolem normal form", are used to denote the modal notions defined in the following paragraphs.

3.3. A modal formula A is in *prenex normal form* iff

(i) A is a positive formula, and

(ii) A is *locally prenex*, i.e. A can no longer be submitted to the transformations used in classical logic to obtain prenex formulae. For example, the following formula is not locally prenex:

$$\forall x(p(x) \vee \exists y \Box(q(x,y) \wedge \forall z \Diamond q(y,z))),$$

because it can still be transformed into

$$\forall x\, \exists y(p(x) \vee \Box \forall z(q(x,y) \wedge \Diamond q(y,z))).$$

This latter formula is locally prenex.

Every formula A can be transformed into an equivalent formula A' which is in prenex normal form.

In what follows, we consider only modal formulae in prenex normal form, so that from now onwards "modal formula" stands, in fact, for "modal formula in prenex normal form". This assumption is not essential for the results obtained here, but it simplifies both the exposition and the proofs.

Moreover, we suppose that if S is a set of formulae, then there are no two bound variables in S with the same name. Finally, we generally speak only of closed formulae. Exceptions are explicitly stated.

The notion of Skolem normal form, which we are now going to define, is somewhat atypical. In fact, if A is a modal formula, its Skolem normal form is an expression which is not a modal formula itself: it is a modal expression, without quantifiers, which can contain symbols labeled by a natural number, which will be written as an exponent.

The procedure for the elimination of existential and universal quantifiers is similar to the corresponding classical one, but here the modal degree of the quantifier being eliminated is saved in the natural number which labels the functional term or the free variable corresponding to it.

3.4. If C is a modal formula in prenex normal form and n is a natural number, then we define the auxiliary transformation Skn by induction on the subformulae of C as follows:

$$\mathrm{Skn}(P,n)=P, \quad \text{if } P \text{ is a literal,}$$

$$\mathrm{Skn}(A \wedge B,n)=\mathrm{Skn}(A,n) \wedge \mathrm{Skn}(B,n),$$

$$\mathrm{Skn}(A \vee B,n)=\mathrm{Skn}(A,n) \vee \mathrm{Skn}(B,n),$$

$$\mathrm{Skn}(\Box A,n)=\Box\,\mathrm{Skn}(A,n+1),$$

$$\mathrm{Skn}(\Diamond A,n)=\Diamond\,\mathrm{Skn}(A,n+1),$$

$$\mathrm{Skn}(\forall x A(x),n)=\mathrm{Skn}(A(x^n),n),$$

$$\mathrm{Skn}(\exists x A(x),n)=\mathrm{Skn}(A(f_x^n(y_1,\ldots,y_m)),n),$$

where f_x is a new functional symbol, and $y_1,\ldots,y_m$ are all the variables free in $\exists x A(x)$.

We are finally ready to define the modal skolemization analog.

3.5. If C is a modal formula in prenex normal form, then C is transformed into *Skolem normal form* by the application of the transformation Sk defined by

$$\mathrm{Sk}(C)=\mathrm{Skn}(C,0).$$

If $S=\{A_1,\ldots,A_n\}$ is a set of modal formulae in prenex normal form, then $\mathrm{Sk}(S)$ is the set $\{\mathrm{Sk}(A_1),\ldots,\mathrm{Sk}(A_n)\}$.

3.6. If a symbol s in $\mathrm{Sk}(C)$ is labeled by the natural number n, ten n is called the *level* of s in $\mathrm{Sk}(C)$.

It can easily be verified that for any quantified variable x occurring in C the natural number labeling (the outermost symbol of) the corresponding term in $\mathrm{Sk}(C)$ is equal to the modal degree of x in C. Here are some examples:

$$\mathrm{Sk}(\forall x \exists y \,\Box p(x,y)) = \Box p(x^0, f^0(x^0)),$$

$$\mathrm{Sk}(\forall x \,\Box \exists y\, p(x,y)) = \Box p(x^0, f^1(x^0)),$$

$$\mathrm{Sk}(\Box \forall x \,\exists y\, p(x,y)) = \Box p(x^1, f^1(x^1)).$$

In the sequel, the symbols without levels are considered to be in the level 0, i.e. the set of all symbols in level 0 contains also the symbols without levels.

Skolem normal formulae are examples of expressions called labeled formulae, which are defined as follows.

3.7. A modal expression E is a *labeled formula* iff it is obtained from a modal formula without quantifiers (which can, however, contain free variables) by assigning a level to some constant, variable or functional symbol in such a way that

(i) if s is a constant, a variable or a functional symbol, then either every occurrence of s in E is labeled by the same level, or s has no levels;

(ii) if a symbol s is in the level n, then every occurrence of s is in the scope of at least n modal operators;

(iii) if f is a functional symbol in level n, which occurs in E in the form $f(t_1,\ldots,t_m)$, then the level of each symbol occurring in t_i, for all $1\leqslant i\leqslant m$, is less than or equal to n.

For any modal formula C, $\mathrm{Sk}(C)$ clearly satisfies (i)–(iii).

Labeled formulae are denoted by capital letters $(A,B,C,\ldots)$, just like modal formulae. To avoid confusion, we shall always state explicitly whether A $(B,C,\ldots)$ is a labeled formula or a modal formula.

4. Modal substitutions and unification

In this section we are going to define a notion of substitution for labeled formulae, which is similar to the corresponding classical notion, except there are some restrictions imposed on the levels of symbols. What we obtain is in fact a notion which has some similarity with substitution in many-sorted logics: here sorts are denoted by natural numbers, called 'levels', and they are such that the set of objects of sort n is a subset of the set of objects of sort m for each $m>n$. Substitutions are denoted by greek lower case letters $(\theta,\sigma,\mu,\ldots)$.

4.1. A *modal substitution* is a finite set of the form $\{t_1/x_1,\ldots,t_n/x_n\}$, where

(1) every x_i is a variable, every t_i is a term different from x_i, and for all i,j such that $i\neq j$, x_i is different from x_j;

(2) every variable x_i and every symbol occurring in t_i may be labeled by a level;

(3) for all i, if n is the level of x_i, then for every symbol s^m, which occurs in t_i, m is less than or equal to n.

In other words, if the level of x is n and if the term t contains some symbol whose level is greater than n, then the substitution of t for x is forbidden.

If E is an expression, the result of the application of the substitution θ to E is denoted by $E\theta$.

The definitions of *composition of modal substitutions* (denoted by $\theta \circ \sigma$), *modal unifier* of a set of expressions, and *most general modal unifier* of a set of expressions are the same as in classical logic. Obviously, the modal notions obtained are different from the corresponding classical ones. However, as there is no risk of confusion, in the sequel we simply speak of substitutions, unifiers, m.g.u., etc.

Some examples will give an intuitive explication of the third restriction in the definition of substitution. Let us consider the set of modal formulae $S=\{\Box\forall x\, p(x), \exists y \Diamond \neg p(y)\}$, $\mathrm{Sk}(S)=\{\Box p(x^1), \Diamond \neg p(c^0)\}$. As the level of x is greater than the level of c, the substitution of c for x is allowed, thus leading to the inconsistent set $\{\Box p(c^0), \Diamond \neg p(c^0)\}$. The admissibility of the substitution reflects the theorems governing the movement of universal quantifiers across modal operators (Section 2). In fact, $\Box\forall x\, p(x)$ implies $\forall x\, \Box p(x)$, i.e. the universal quantifier in S can be brought to a position where it has the same modal degree as the existential quantifier. Further simple transformations show that $\forall x\, \Box p(x)$ is equivalent to $\neg\exists x \Diamond \neg p(x)$.

Let now S be $\{\forall x\, \Box p(x), \Diamond\exists y \neg p(y)\}$. Here none of the two quantifiers can be moved. This fact is respected by the definition of substitution, which does not allow the replacement of x by c in $\mathrm{Sk}(S)=\{\Box p(x^0), \Diamond \neg p(c^1)\}$.

It should be pointed out that, unlike [21], the notion of modal unification does not answer for sets of modal formulae which are consistent by virtue of the propositional form of their components. For example, let S be the set $\{\Diamond\forall x p(x), \exists y \Diamond \neg p(y)\}$ and $\mathrm{Sk}(S)=\{\Diamond p(x^1), \Diamond \neg p(c^0)\}$. In $\mathrm{Sk}(S)$, x would be replaceable with c, but, as we shall see in the next section, it is the propositional basis of the system that prevents the application of the resolution rule.

5. The resolution rules

The resolution system is here defined in detail only for the system D, even if some hints are given also for the systems T and $S4$. The same will be done in the completeness and soundness proofs.

Resolution rules are defined over labeled formulae in such a way that a set S of closed first-order modal formulae (in prenex normal form) is contradictory if and only if $\mathrm{Sk}(S)$ is refutable by means of the resolution rules. So, in order to refute S, each formula in S is first transformed into Skolem normal form; then the resolution rules can be applied.

The modal first-order resolution system is defined by merging the notion of modal substitution and the propositional resolution rules for the system D defined in [15] or [12]. However, the structure of the definition given here is slightly different from [15] and [12], because the rules can here be applied to formulae which need not be in modal clausal form.

In what follows, $\perp$ denotes the empty formula (or the false).

5.1. Let S be a set containing one or two labeled formulae and θ a substitution. Then C is a *θ-resolvent* of S if it can be obtained by applying the following recursive definition to the subformulae of S. The definition is given by three sets of rules as follows, where we use S' to denote a set containing at most a single formula.

5.1.1. Classical rules

(c1) $\perp$ is a θ-resolvent of $\{P_1, \neg P_2\}$ if θ is an m.g.u. of $\{P_1, P_2\}$.
(c2) If C is a θ-resolvent of $S' \cup \{A\}$, then $C \vee B\theta$ is a θ-resolvent of $S' \cup \{A \vee B\}$.
(c3) If C is a θ-resolvent of $S' \cup \{A\}$, then $C \wedge B\theta$ is a θ-resolvent of $S' \cup \{A \wedge B\}$.
(c4) If C is a θ-resolvent of $\{A, B\}$, then C is a θ-resolvent of $\{A \wedge B\}$.

5.1.2. Modal rules

(m1) If C is a θ-resolvent of $\{A, B\}$, then $\Box C$ is a θ-resolvent of $\{\Box A, \Box B\}$.
(m2) If C is a θ-resolvent of $\{A\}$, then $\Box C$ is a θ-resolvent of $\{\Box A\}$.
(m3) If C is a θ-resolvent of $\{A, B\}$, then $\Diamond C$ is a θ-resolvent of $\{\Box A, \Diamond B\}$.
(m4) If C is a θ-resolvent of $\{A\}$, then $\Diamond C$ is a θ-resolvent of $\{\Diamond A\}$.

5.1.3. Simplification rules

(s1) If $\perp \wedge A$ is a θ-resolvent of S, then $\perp$ is a θ-resolvent of S.
(s2) If $\perp \vee A$ is a θ-resolvent of S, then A is a θ-resolvent of S.
(s3) If $\Box \perp$ is a θ-resolvent of S, then $\perp$ is a θ-resolvent of S.
(s4) If $\Diamond \perp$ is a θ-resolvent of S, then $\perp$ is a θ-resolvent of S.

For example, a resolvent of the set of labeled formulas $\{\Diamond(p(x^1) \wedge q(x^1)), \Box(\neg p(c^0) \vee r(c^0))\}$ can be constructed starting from the fact that $\perp$ is a θ-resolvent of $\{p(x^1), \neg p(c^0)\}$ for $\theta = \{c^0/x^1\}$ by rule c1. From this follows by rule c2 that $\perp \vee r(c^0)$ is a θ-resolvent of $\{p(x^1), \neg p(c^0) \vee r(c^0)\}$, and by s2 that $r(c^0)$ is a θ-resolvent of $\{p(x^1), \neg p(c^0) \vee r(c^0)\}$. By c3, $r(c^0) \wedge q(c^0)$ is a θ-resolvent of $\{p(x^1) \wedge q(x^1), \neg p(c^0) \vee r(c^0)\}$, and, finally, $\Diamond(r(c^0) \wedge q(c^0))$ is a θ-resolvent of $\{\Diamond(p(x^1) \wedge q(x^1)), \Box(\neg p(c^0) \vee r(c^0))\}$.

This example shows also that the θ-resolvent of a set of formulae is not uniquely determined. In fact, if the above sequence of applications of rules c2, s2, c3 is replaced by the sequence c3, s1, c2, s2, then we obtain that $\Diamond r(c^0)$ is a θ-resolvent of $\{\Diamond(p(x^1) \wedge q(x^1)), \Box(\neg p(c^0) \vee r(c^0))\}$.

To simplify the proofs of the theorems, from now onwards $\Box\bot$ and $\Diamond\bot$ are considered identical to $\bot$, so that the rules s3 and s4 can be omitted from the definition above.

The rules for system D are of the simplest kind, because D does not include laws that may change the modal degree of formulae such as

(T) $\Box A \to A$;

($S4$) $\Box A \to \Box\Box A$.

In both (T) and ($S4$), the modal degree of A in the antecedent of the implication is different from the modal degree of A in the consequent. The definition of the resolution rule for the systems T and $S4$ must contain some additional clauses which reflect the possibility of deriving $\mathrm{Sk}(A)$ from $\mathrm{Sk}(\Box A)$, and $\mathrm{Sk}(\Box\Box A)$ from $\mathrm{Sk}(\Box A)$. Here the level of terms must be manipulated according to the modification of the modal degrees of the corresponding variables in the modal formulae.

Thus, to obtain a resolution system for the systems T and $S4$, before starting the application of the recursive definition of θ-resolvent, the modal degree of subformulae in S should be remembered. The following items should then be added to the modal rules:

(m5) If C is a θ-resolvent of $S' \cup \{A'\}$, then C is a θ-resolvent of $S' \cup \{\Box A\}$, where A' is obtained from A by subtracting one from the level of those symbols which have a level greater than the modal degree of $\Box A$ in S.

and, for the system $S4$ only

(m6) If C is a θ-resolvent of $S' \cup \{\Box\Box A'\}$, then C is a θ-resolvent of $S' \cup \{\Box A\}$, where A' is obtained from A by adding one to the level of those symbols which have a level greater than the modal degree of $\Box A$ in S.

5.2. The resolution system consists of the following rules (the first two rules are structural rules, while the third is the resolution rule).

Factorization rule:

$$\frac{C[D \vee F]}{(C[D])\theta}$$

if θ is an m.g.u. of D and F.

Duplication rule:

$$\frac{C[D(x^n)]}{C[D(x^n) \wedge D(y^n)]}$$

if y is a new variable, x^n occurs only in $D(x^n)$, $D(x^n)$ is not in the scope of more than n modal operators, and $D(x^n)$ is not in the scope of a negation.

Resolution rule:

$$\frac{S}{C}$$

if S is a set containing one or two labeled formulae and C is a θ-resolvent of S.

5.3. A *deduction* of a labeled formula A from a set S of labeled formulae is a tree whose root is A, whose leaves are elements of S, and any of its nodes B, whose immediate descendent(s) is (are) C (and D) is the conclusion of one of the inference rules given above, whose premiss (premisses) is (are) C (and D).

5.4. A deduction of the empty formula $\perp$ from the set S of labeled formulae is a *refutation* of S. If there exists a refutation of S, then S is called *refutable.*

We give here a sample refutation of a set of formulae. Let

$$C_1 = \exists x_1 \Box \forall x_2 \Box (p(x_1) \wedge \neg q(x_1, x_2)),$$

$$C_2 = \exists y_1 \Box \forall y_2 (\neg r(y_1, y_2) \vee \forall y_3 \Diamond q(y_3, y_2)),$$

$$C_3 = \forall z_1 \Diamond \exists z_2 (r(z_1, z_2) \vee \forall z_3 \Box \neg p(z_3)).$$

The set $S = \{C_1, C_2, C_3\}$ is contradictory.

The set $\mathrm{Sk}(S) = \{\mathrm{Sk}(C_1), \mathrm{Sk}(C_2), \mathrm{Sk}(C_3)\}$ is constructed as follows:

$$\mathrm{Sk}(C_1) = \Box\Box(p(c^0) \wedge \neg q(c^0, x_2^1)),$$

$$\mathrm{Sk}(C_2) = \Box(\neg r(d^0, y_2^1) \vee \Diamond q(y_3^1, y_2^1)),$$

$$\mathrm{Sk}(C_3) = \Diamond(r(z_1^0, f^1(z_1^0)) \vee \Box \neg p(z_3^1)),$$

and the following is a refutation of $\mathrm{Sk}(S)$:

$$
{\scriptstyle \{c/y_3,\, f(d)/x_2\}}\;\dfrac{{\scriptstyle \{d/z_1,\, f(d)/y_2\}}\;\dfrac{{\scriptstyle \{c/z_3\}}\;\dfrac{\Box\Box(p(c^0) \wedge \neg q(c^0, x_2^1)) \quad \Diamond(r(z_1^0, f^1(z_1^0)) \vee \Box \neg p(z_3^1))}{\Diamond r(z_1^0, f^1(z_1^0))} \qquad \Box(\neg r(d^0, y_2^1) \vee \Diamond q(y_3^1, y_2^1))}{\Diamond\Diamond q(y_3^1, f^1(d^0))} \qquad \Box\Box(p(c^0) \wedge \neg q(c^0, x_2^1))}{\perp}
$$

The substitution applied to obtain each resolvent in the derivation is shown next to the corresponding derivation line (the levels of the symbols have been omitted).

6. Soundness and completeness

The soundness and completeness proofs are obtained by means of two fundamental theorems. The first is a version of Herbrand's property, limited to those quantifiers which are in the scope of no modal operator. In other words, this theorem reduces the contradictority of a set of first-order modal formulae to the contradictority of a set of formulae, which contain no quantifier outside the scope of modal operators. We also state a different version of the theorem, which holds for the systems T and $S4$. The formulation of a general Herbrand's property for the modal system D can be found in [8] or [11]. An approach to the same problem for the systems T and $S4$ can be found in [10].

The second theorem is a first-order extension of the result obtained in [15], which reduces the contradictority of a set of formulae whose main logical symbol is a modal operator, to the contradictority of a set of formulae which contain less modal operators. The propositional versions of this theorem for the systems T and $S4$, whose formulation can be found in [10], is implicitly contained in [12].

By the repeated application of either result to a set S of modal formulae, quantifiers and modal operators can be eliminated, thus reducing the contradictority of S to the contradictority of a set of propositional (classical) formulae.

6.1. In order to state the Herbrand theorem analog, we give some definitions which allow the construction of a kind of functional form of formulae. They are essentially the same as the corresponding ones in classical logic, except that in the case of modal logic they are restricted to quantifiers whose modal degree is 0 (quantifiers which are in the scope of no modal operator). In other words, the functional form of modal formulae is the result of the application of the classical method for the elimination of quantifiers to quantifiers of modal degree 0 only.

6.1.1. If S is a set of modal formulae and $A \in S$, then $E(A)$ is the formula obtained from A by deleting every existential quantifier $\exists x$ whose modal degree is 0 and replacing every occurrence of x by its *functional term*, which has the following form:

$$f_x(y_1, \ldots, y_n),$$

where f_x is a Skolem function for x, and $y_1, \ldots, y_n$ are all the universal variables such that $\exists x$ is in the scope of $\forall y_i$.

We define $E(S) = \{E(A)\colon A \in S\}$.

6.1.2. Let S be a set of formulae. The sequence of the *H-domains of level* 0 for $S, H_i(S)$ is defined as follows, where a is an arbitrary constant which does not occur in S:

$$H_0(S) = \{a\} \cup \{c\colon c \text{ is a constant which occurs in } S\},$$

$$H_{i+1}(S) = H_i(S) \cup \{f(t_1, \ldots, t_n)\colon f \text{ is a functional symbol which occurs in } S \text{ and } t_1, \ldots, t_n \in H_i(S)\}.$$

6.1.3. Let S be a set of formulae and $A \in E(S)$. Then A does not contain any existential quantifier whose modal degree is 0. The *jth expansion in level* 0 *of* A, $F_j(A)$, is defined by induction on the number of propositional connectives and quantifiers which are in the scope of no modal operator:

(a) $F_j(P) = P$, if P is a literal;
$F_j(\Box A) = \Box A$;
$F_j(\Diamond A) = \Diamond A$.

(b) $F_j(A \wedge B) = F_j(A) \wedge F_j(B)$.

(c) $F_j(A \vee B) = F_j(A) \vee F_j(B)$.

(d) $F_j(\forall x A(x)) = \bigwedge_{t \in H_j(E(S))} F_j(A(t))$.

We note that clause (d) is applied only if the modal degree of $\forall x$ is 0. If $E(S)=\{A_1,\dots,A_n\}$, then $F_j(E(S))=\{F_j(A_1),\dots,F_j(A_n)\}$.

6.2. We are now ready to state the Herbrand theorem analog and the propositional fundamental theorem used to prove soundness and completeness of modal first-order resolution for system D.

Theorem 1 (Modal Herbrand's Property for level 0). *A set S of closed modal formulae is contradictory iff for some j, $F_j(E(S))$ is contradictory.*

Theorem 2. *Let S be a set of closed modal formulae of the form*

$$\{\Box A_1,\dots,\Box A_n,\Diamond B_1,\dots,\Diamond B_m,P_1,\dots,P_k\},$$

where $P_1,\dots,P_k$ are literals. S is contradictory iff one of the following sets is contradictory:

$$S_0=\{P_1,\dots,P_k\},$$

$$S_j=\{A_1,\dots,A_n,B_j\},\quad 1\leqslant j\leqslant m.$$

For the soundness and completeness proofs, we need also the following theorem.

Theorem 3. *Let S be a set of closed modal formulae of the form*

$$\{C_1\vee C_2,A_1,\dots,A_n\}.$$

S is contradictory iff the following two sets are both contradictory:

$$S_1=\{C_1,A_1,\dots,A_n\},\qquad S_2=\{C_2,A_1,\dots,A_n\}.$$

It should be noted that if Theorem 1 is applied to sets of formulae without any modal operator, then it is equivalent to the semantical version of Herbrand's theorem for classical logic.

Theorem 1 can be proved semantically along the lines of [8] or [11]. For those modal systems which allow a formalization in terms of sequent calculus, where the cut elimination theorem holds [28], a syntactical proof can be obtained by induction on cut-free proofs. For example, the following version of Theorem 1 holds both for the modal system T and for the system $S4$ (a proof can be found in [10]).

Theorem 1*. *A set $S=\{C_1,\dots,C_n\}$ of closed formulae is contradictory iff for some j, $F_j(E(S^*))$ is contradictory, where $S^*=\{C_1^*,\dots,C_n^*\}$ is the closure of S with respect to the axiom (T), defined for each formula C_i in S by the following recursive clauses:*

$L^*=L$	*if L is a literal,*
$(A\#B)^*=A^*\#B^*$	*if $\#$ is $\wedge$ or $\vee$*
$(QxA)^*=QxA^*$	*if Q is a quantifier,*
$(\Diamond A)^*=\Diamond A^*$,	
$(\Box A)^*=\Box A^*\wedge A^*$.	

6.3. Given a set S of closed modal formulae, an AND/OR tree of nodes labeled by sets of formulae, Tree(S), can be constructed in the following way, where a set of the form $\{C_1 \wedge C_2, A_1, \ldots, A_n\}$ is considered the same as $\{C_1, C_2, A_1, \ldots, A_n\}$. It will be noticed that Tree(S) is very close to the tree constructed using the standard tableaux methods:

(1) The root of Tree(S) is labeled by S.

(2) The immediate descendents of a node i of the tree, labeled by the set S_i, are OR nodes constructed in accordance with the following rules:

(a) If S_i contains some quantifier whose modal degree is 0, then i has (possibly an infinite number of) immediate descendents, labeled by the sets

$$F_0(E(S_i)), F_1(E(S_i)), F_2(E(S_i)), \ldots$$

Otherwise:

(b1) If $S_i = \{C_1 \vee C_2, A_1, \ldots, A_n\}$, then one of the immediate descendents of i is a node without label, which has exactly two AND nodes as immediate descendents, labeled by

$$\{C_1, A_1, \ldots, A_n\} \quad \text{and} \quad \{C_2, A_1, \ldots, A_n\}.$$

(b2) If S_i has the form

$$\{\Box A_1, \ldots, \Box A_n, \Diamond B_1, \ldots, \Diamond B_m, P_1, \ldots, P_k\},$$

then the following sets are labels of the immediate descendents of i:

$$\{P_1, \ldots, P_k\},$$

$$\{A_1, \ldots, A_n, B_j\} \quad \text{for } j = 1, \ldots, m.$$

To state the next result, we need the following definitions.

6.3.1. Let S be a set of first-order modal formulae. A node i of Tree(S), labeled by S_i, is called *closed* iff one of the following conditions holds:

(i) i is a terminal node and for some atom P, $P \in S_i$ and $\neg P \in S_i$;

(ii) i is an AND node and every immediate descendent of i is closed;

(iii) i is an OR node and at least one of the immediate descendents of i is closed.

6.3.2. If S is a set of modal formulae, then Tree(S) is called closed iff its root is closed.

Theorems 1–3 allow to prove the following fundamental result.

Theorem 4. *A set S of first-order modal formulae is contradictory iff* Tree(S) *is closed.*

Finally, this theorem is used to prove the completeness and soundness of the resolution method.

Theorem 5 (Completeness and soundness theorem). *A set S of first-order modal formulae is contradictory iff* Sk(S) *is refutable.*

Theorem 5 can be proved along the following lines. By Theorem 4, it is sufficient to show that Tree(S) is closed iff Sk(S) is refutable. The proof of this latter assertion is by induction on the depth of Tree(S), and it amounts to showing that for any set S of closed modal formulae,

(i) if S is the label of a terminal node of Tree (S') for some set S' of modal formulae, then Sk(S) is refutable iff S contains both P and $\neg P$ for some atom P;

(ii) Sk(S) is refutable iff for some j, Sk($F_j(E(S))$) is refutable.

(iii) If $S=\{\Box A_1,\ldots,\Box A_n, \Diamond B_1,\ldots,\Diamond B_m, P_1,\ldots,P_k\}$, where $P_1,\ldots,P_k$ are literals, then Sk(S) is refutable iff one of the sets Sk(S_j), for $1\leqslant j\leqslant m$, is refutable, where

$$S_0=\{P_1,\ldots,P_k\}, \qquad S_j=\{A_1,\ldots,A_n,B_j\}.$$

(iv) If $S=\{C_1 \vee C_2, A_1,\ldots,A_n\}$, then Sk($S$) is refutable iff the sets Sk(S_1) and Sk(S_2) are both refutable, where

$$S_1=\{C_1,A_1,\ldots,A_n\}, \qquad S_2=\{C_2,A_1,\ldots,A_n\}.$$

A detailed proof of these assertions can be found in [9].

7. Related work

Konolige [23] and Geissler and Konolige [19] propose a resolution method for modal logics of knowledge and belief whose propositional basis is the reduction theorems similar to our Theorem 2 in Section 6.2 and the corresponding theorems for other modal theories ([10, 12]). However, their recursive application is made by explicitly calling the theorem prover itself to determine whether two formulae can be resolved against each other. This mechanism may give rise to serious complexity problems in a possible implementation of the system.

As far as the quantificational aspects of Konolige's system is concerned, it should be pointed out that the main problem arises here from the fact that Konolige considers modal logics whose semantics allow different individuals to be the interpretation of the same term in different possible worlds. So, a term occurring in a modal context may denote either its interpretation in the actual world (rigid designation), or what one believes its designation to be (nonrigid designation).

The method of recursively calling the theorem prover, after dropping some external modal operator, leaves only the problem of determining unifiability of terms which have eventually been brought outside any modal context. In fact, modal formulae are considered as unanalyzed predications and "internal" quantifiers are not taken into account until the recursive calls to the theorem prover have brought them to the surface. The difference between rigid and nonrigid designation is treated by means of a "bullet constructor", which turns nonrigid terms into rigid ones when they are in a modal context, i.e. a term in the scope of the rigid designation operator always denotes its interpretation in the actual world. The bullet constructor guarantees the correct elimination of quantifiers and the soundness of the unification procedure.

Abadi and Manna [2] define nonclausal resolution proof systems for several modal logics. Some of the rules of the systems are rewrite rules which can introduce new logical operators, and whose aim is the transformation of formulae so as to bring possible inconsistencies, i.e. complementary subformulae, into the same "possible world", i.e. modal context. Only then can resolution be applied. Should the system be fully automatized, such chains of steps would risk to increase the length of the procedure redundantly, unless suitable heuristics are implemented.

Abadi and Manna's resolution method manipulates formulae with quantifiers. Although their approach is very stimulating, it suffers somehow from the same limitations as the propositional basis: transformations of formulae may be necessary before the resolution rule can be applied to two quantified formulae, and such steps may eventually turn out to be useless. In fact, it does not seem easy to recognize from outside (i.e. before exploring all the needed transformations of formulae) whether two expressions will eventually show to be unifiable or not, in order to decide whether an attempt to apply the resolution rule might be worthwhile.

The authors propose also a modal version of skolemization, although their resolution method does not rely upon it. It is in fact presented only as a sometimes-useful short-cut. Soundness of skolemization is guaranteed by the use of nonrigid symbols (thus, the modal language is extended with the distinction between rigid and nonrigid symbols), and its completeness is due to the possible introduction of equations in skolemized formulae. This seems to be a very high price to pay.

Jackson and Reichgelt [21] present a proof method for first-order modal logics based on sequent calculus. Modality is represented by an indexing of formulae which reflects the path leading from the actual world to the world where they are true. Dependencies on worlds are thus made perfectly explicit. The quantifier elimination rules introduce Skolem functions whose arguments may be individual variables as well as world variables. The differences among modal systems are entirely embedded in the nature of the unification algorithm, whose definition is conditioned by the features of the accessibility relation for a given logic. We believe that the explicit reference to Kripke semantics, although intuitive, makes it hard to think of a full automation of the system.

8. Concluding remarks

In this work we presented a resolution method for quantified modal logics. The propositional method underlying our system is a minor modification of the systems proposed by Fariñas in [12, 13, 15].

If we wish to compare Fariñas's approach to the methods briefly discussed in the latter section, we remark that it seems to keep modal resolution closer to the classical method. Obviously, modal operators introduce a considerable loss of simplicity, but the use of "single-step" resolution rules allows an early elimination of complementary literals. Another important remark should be made in favour of this method: it can be adapted to formulae which are in a simple normal form, close to the classical clausal form, and refined so as to obtain linear deductions [17]. It is this result which allows the definition of the modal logic programming language Molog, which has been implemented and shown to have a clear declarative semantics [5].

The unification procedure needed to extend resolution systems acting only on prenex formulae to full first-order logic does not introduce any test except for simple numeric tests (greater-than tests on the level of symbols), so that the extension of Molog to the case of formulae with quantifiers in the scope of modal operators should not be of great difficulty. The possibility to check whether two expressions are unifiable before applying the recursive definition of modal resolvent maintains the validity of the strategies adopted by Molog to decide whether or not to attempt an application of the resolution rule and to avoid useless applications of rules which increase the modal complexity of formulae (such as rule m6 for system *S4* in Section 5.1).

We believe that our approach to unification in modal contexts should be accessible also from different propositional proof procedures for modal logics, such as sequent-based or tableaux-like systems.

References

[1] M. Abadi and Z. Manna, Non clausal temporal deduction, in: R. Parikh, ed., *Proc. of the Logics of Programs Conference*, Lecture Notes in Computer Science, Vol. 193 (Springer, Berlin, 1985) 1–15.

[2] M. Abadi and Z. Manna, Modal theorem proving, in: *Proc. of the 8th International Conference on Automated Deduction*, Lecture Notes in Computer Science Vol. 230 (Springer, Berlin, 1986), 172–189.

[3] M. Abadi and Z. Manna, A timely resolution, in: *Proc. of the Symposium on Logic and Computer Science*, Cambridge MA, (1986), 176–186.

[4] R. Arthaud, P. Bieber, L. Fariñas del Cerro, J. Henry and A. Herzig, MOLOG-Manuel d'utilization, Report Université Paul Sabatier, Toulouse, 1986.

[5] P. Balbiani and L. Fariñas del Cerro, Declarative Semantics for Modal Logic Programs, Report Université Paul Sabatier, Toulouse, 1987.

[6] M. Bricard, Une machine abstraite pour compiler MOLOG, Report Université Paul Sabatier, Toulouse, 1987.

[7] C. Chang and R. Lee, *Symbolic Logic and Mechanical Theorem Proving* (Academic Press, New York, 1973).

[8] M. Cialdea, Une méthode de déduction automatique en logique modale, Thesis, Université Paul Sabatier, Toulouse, 1986.

[9] M. Cialdea, Modal first-order resolution, Report Université Paul Sabatier, Toulouse, 1987.

[10] M. Cialdea, Towards first-order resolution for the modal systems T and S4, Report Université Paul Sabatier, Toulouse, 1988.

[11] M. Cialdea and L. Farinãs del Cerro, A modal Herbrand's property, *Z. Math. Logik Grundlag. Math.* **32** (1986) 523–530.

[12] P. Enjalbert and L. Fariñas del Cerro, Modal resolution in clausal form, Report Greco Programmation, R. G. 14–86, 1986 *Theoret. Comput. Sci.* **65** (1989) 1–34.

[13] L. Fariñas del Cerro, A simple deduction method for modal logic, *Inform. Process. Lett.* **14** (1982) 49–51.

[14] L. Fariñas del Cerro, Temporal reasoning and termination of programs, in: *Proc. of the 8th Internat. Joint Conf. on Artificial Intelligence*, Karlsruhe, West Germany (1983) 926–929.
[15] L. Fariñas del Cerro, Resolution modal logic, *Logique et Analyse* **28** (1985) 153–172.
[16] L. Fariñas del Cerro, MOLOG, a system that extends PROLOG with modal logic, *The New Generation Computer Journal* **4** (1986) 35–51.
[17] L. Fariñas del Cerro and A. Herzig, Linear modal deductions, Report Université Paul Sabatier, Toulouse, 1987.
[18] L. Fariñas del Cerro and E. Orlowska, eds., Special Issue on Automated reasoning in nonclassical logic, *Logique et Analyse* **28** (1985).
[19] C. Geissler and K. Konolige, A resolution method for quantified modal logics of knowledge and belief, in: *Proc. of the Conf. on Theoretical Aspects of Reasoning about Knowledge*, Monterey, California (1986) 309–324.
[20] G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic* (Methuen & Co., London, 1968).
[21] P. Jackson and H. Reichgelt, A general proof method for first-order modal logic, in: *Proc. of the 10th Internat. Joint Conf. on Artificial Intelligence*, Morgan Kaumann, Los Altos, California (1987) 942–944.
[22] K. Konolige, A deduction model of belief and its logic, SRI International, Technical Note 326, 1984, Menlo Park, California. (A revised version appeared as *A deduction model of belief*, Pitman, London, 1986).
[23] K. Konolige, Resolution and quantified epistemic logics, in: *8th Internat. Conf. on Automated Deduction*, Lecture Notes in Computer Science Vol. 230 (Springer, Berlin, 1986) 199–208.
[24] D. Lehmann, Knowledge, common knowledge and related puzzles, in: *Proc. of the 3rd ACM Conf. on Principle of Distributed Computing*, 1984, 62–67.
[25] Z. Manna and A. Pnueli, The modal logic of programs, in: *Internat. Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science Vol. 71 (Springer, Berlin, 1979).
[26] Z. Manna and A. Pnueli, Verification of concurrent programs: the temporal framework, in: R.S. Boyer, J.S. Moore, eds., *The Correctness Problem in Computer Science* (Academic Press, London, 1982) 215–273.
[27] Z. Manna and P. Wolper, Synthesis of communicating processes from temporal logic specifications, *ACM Trans. on Programming Languages and Systems* **6** (1984) 68–93.
[28] M. Ohnishi and K. Matsumoto, Gentzen method in modal calculi, *Osaka Mathematical Journal* **9** (1957) 113–130.
[29] E. Orlowska and Z. Pawlak, Expressive power of knowledge representation systems, ICSFAS Report 432, 1981.
[30] J. Robinson, A machine oriented logic based on the resolution principle, *J. ACM* **12** (1965) 23–41.